1) Pretrained Model for Image Detection

For the task of image detection using the Galaxy Zoo dataset, a suitable pretrained model can be found on Hugging Face, which offers a range of models for image-related tasks. Here are some options:

- **Vision Transformers (ViT)**: This model performs well on image classification tasks and can be fine-tuned on specific datasets. It's efficient and often used in astronomy-related projects due to its ability to handle large image datasets.

Why ViT?

- Good for tasks like galaxy classification due to its focus on image patches, which can capture intricate spatial relationships in the images.
- The model is relatively lightweight compared to some CNNs and can run feasibly on most modern hardware.

Framework:

- Hugging Face offers a PyTorch implementation, and you can use the transformers library along with datasets and accelerate to streamline fine-tuning and inference.

Other Model Options:

- **EfficientNet**: Known for its efficiency in terms of performance and resource consumption.
- **ResNet**: A classic model with strong baseline performance, especially ResNet50 or ResNet101 for more complex tasks.

Resource:Already worked on these models before and it give efficient results for image classification

2) Fine-tuning Methods

- **Full Fine-Tuning**:
  - This method involves updating all layers of the pretrained model on the new dataset. It's most effective when the new dataset has substantial differences from the original dataset used for training the model.
  - Suitable for Galaxy Zoo due to the unique nature of astronomical images.

Else we can do

Transer learning i.e.

- **Layer-Freezing & Fine-Tuning**:
  - You can freeze the early layers of the model (responsible for low-level feature extraction) and only fine-tune the later layers.
  - This method reduces training time and computational costs while preserving the general knowledge in earlier layers.

- **Learning Rate Scheduling**:
  - Use discriminative learning rates, where different parts of the model are fine-tuned at different learning rates. The earlier layers get smaller learning rates, and later layers get higher ones.

Framework for Fine-Tuning:

- Use Hugging Face's Trainer API to handle the training loop, evaluation, and optimizer configuration efficiently.
- Use torch.optim for custom optimizers if you prefer finer control.

3) Ensemble Methods (Optional)

Ensemble methods can potentially boost the performance beyond standard fine-tuning techniques. Here are some options:

- **Model Averaging**: Train multiple models (e.g., ViT, ResNet) and average their predictions to create a more robust output.
- **Stacking**: You can combine the outputs of several models using a meta-model to learn the best way to combine the predictions from each model.
- **Cross-Domain Ensembles**: Combine models trained on different types of image data (e.g., one on natural images, another on astronomical images) to enhance the detection and classification accuracy for edge cases in the Galaxy Zoo dataset.

Resource: https://www.v7labs.com/blog/ensemble-learning
And 100 days of machine learning